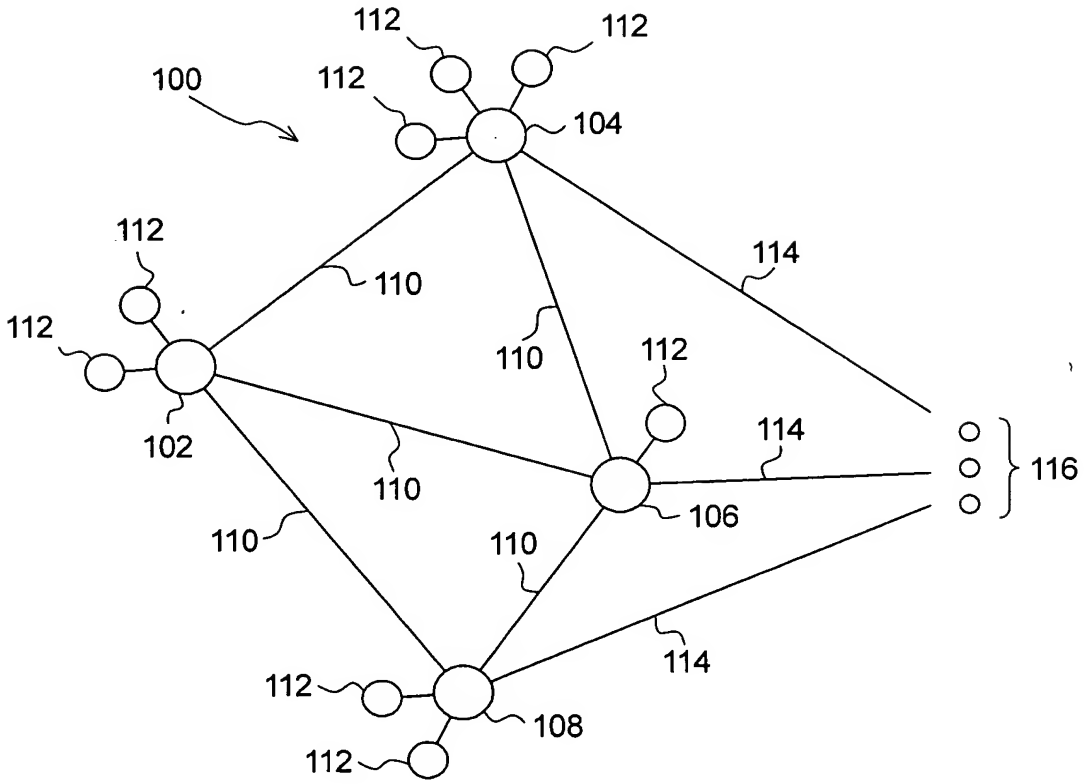
100
102
104
106
108
110
112
114
116

FIG. 1

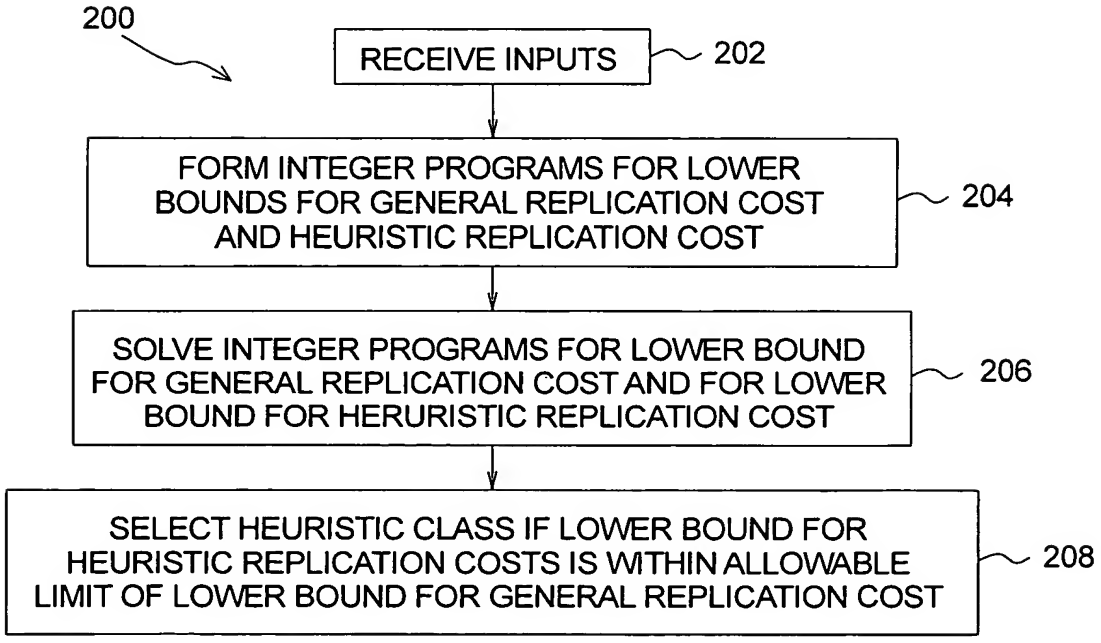
200
RECEIVE INPUTS
202
FORM INTEGER PROGRAMS FOR LOWER BOUNDS FOR GENERAL REPLICATION COST AND HEURISTIC REPLICATION COST
204
SOLVE INTEGER PROGRAMS FOR LOWER BOUND FOR GENERAL REPLICATION COST AND FOR LOWER BOUND FOR HERURISTIC REPLICATION COST
206
SELECT HEURISTIC CLASS IF LOWER BOUND FOR HEURISTIC REPLICATION COSTS IS WITHIN ALLOWABLE LIMIT OF LOWER BOUND FOR GENERAL REPLICATION COST
208

FIG. 2

Table 1		
Variable	Type	Description
$store_{nik}$	Binary	nth node stores kth data object during ith interval
$create_{nik}$	Binary	kth data object created on nth node during ith interval
$covered_{nik}$	Binary	nth node can access kth data object w/in latency threshold during ith interval
$open_{n}$	Binary	nth node is enabled; it can store replicas and run data placement heuristics

FIG. 3

Table 2		
Variable	Type	Description
$read_{nik}$	Integer	Number of read requests from nth node for kth data object during ith interval
$write_{nik}$	Integer	Number of write requests from nth node for kth data object during ith interval
$latency_{nm}$	Real	Data access latency for accessing an object ath mth node from nth node
$dist_{nm}$	Binary	nth node can access mth node within latency threshold
$know_{nm}$	Binary	nth node uses information from mth node to make placement decision
$fetch_{nm}$	Binary	nth node can fetch data objects form mth node
$hist_{nik}$	Binary	nth node has a record in its activity history of kth data object being accessed on nth node during ith interval
T_{lat}	Real	Target latency threshold
T_{qos}	Real	Specified ratio of accesses served within latency threshold
α	Integer	Unit cost for storage
β	Integer	Unit cost for replica creation
γ	Integer	Unit cost penalty for an access exceeding latency threshold
δ	Integer	Unit cost for an update message
ζ	Integer	Unit cost for enabling and managing a node

FIG. 4

Table 3						
Heuristic class	Heuristic properties					
	Storage constraint	Replica constraint	*route*	*know*	*hist*	Reactive constraint
Storage constrained heuristics	x		global	global	multi	
Replica constrained heurisitics		x	global	global	multi	
Decentralized storage constrained heuristics w/ local routing	x		local	local	multi	
Local caching	x		local	local	single	x
Cooperative caching	x		global	global	single	x
Local caching w/ prefetching	x		local	local	single	
Cooperative caching w/ prefetching	x		global	global	single	

FIG. 5

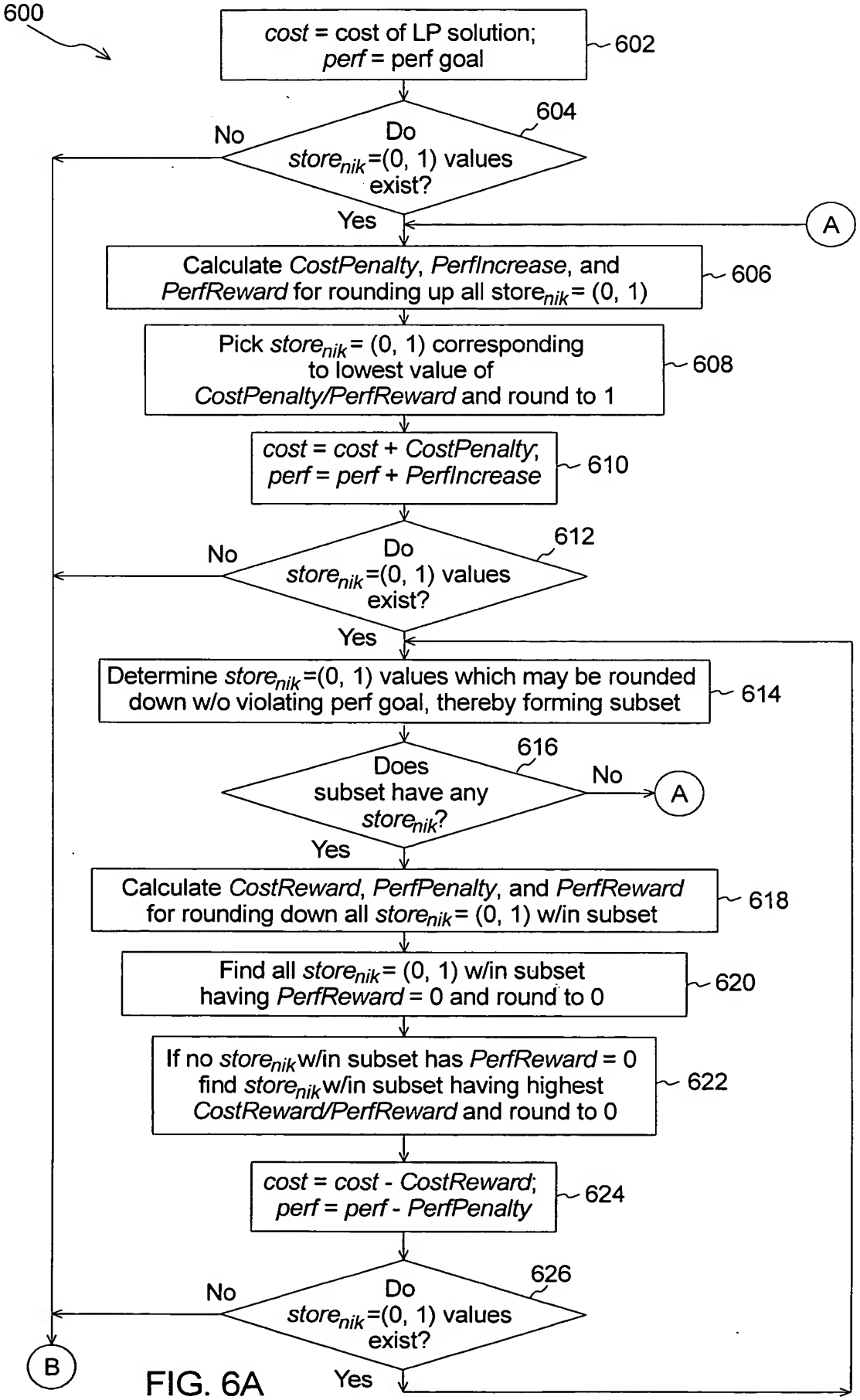
600
cost = cost of LP solution;
perf = perf goal
602
Do
store_nik=(0, 1) values
exist?
604
No
Yes
A
Calculate CostPenalty, PerfIncrease, and
PerfReward for rounding up all store_nik= (0, 1)
606
Pick store_nik= (0, 1) corresponding
to lowest value of
CostPenalty/PerfReward and round to 1
608
cost = cost + CostPenalty;
perf = perf + PerfIncrease
610
Do
store_nik=(0, 1) values
exist?
612
No
Yes
Determine store_nik=(0, 1) values which may be rounded
down w/o violating perf goal, thereby forming subset
614
Does
subset have any
store_nik?
616
No
A
Yes
Calculate CostReward, PerfPenalty, and PerfReward
for rounding down all store_nik= (0, 1) w/in subset
618
Find all store_nik= (0, 1) w/in subset
having PerfReward = 0 and round to 0
620
If no store_nik w/in subset has PerfReward = 0
find store_nik w/in subset having highest
CostReward/PerfReward and round to 0
622
cost = cost - CostReward;
perf = perf - PerfPenalty
624
Do
store_nik=(0, 1) values
exist?
626
No
Yes
B

FIG. 6A

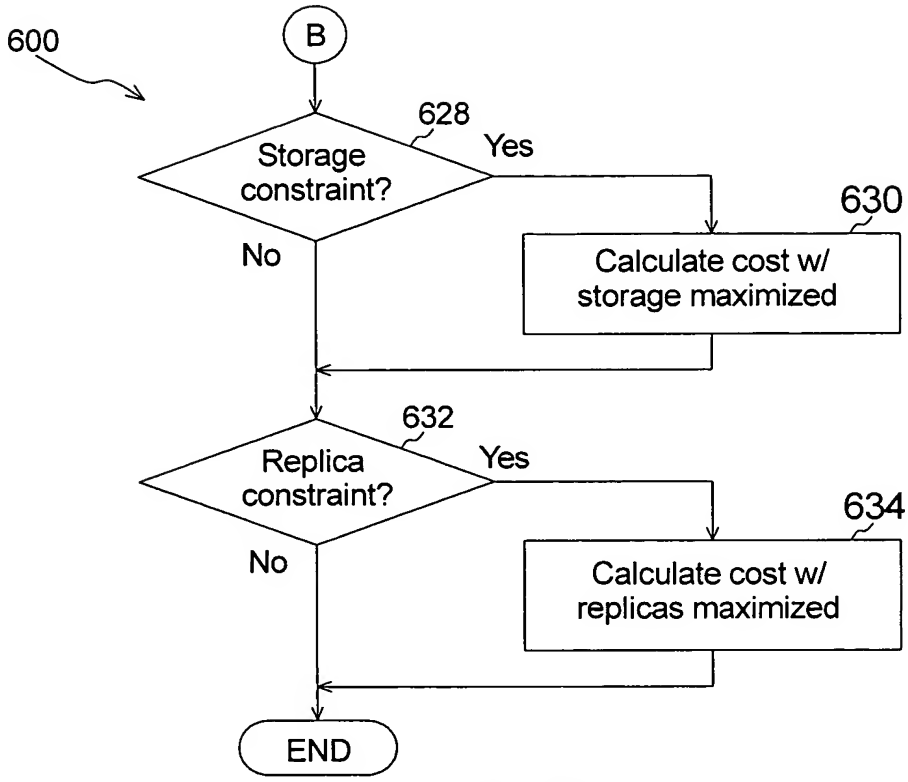
600
B
628
Storage constraint?
Yes
No
630
Calculate cost w/ storage maximized
632
Replica constraint?
Yes
No
634
Calculate cost w/ replicas maximized
END

FIG. 6B

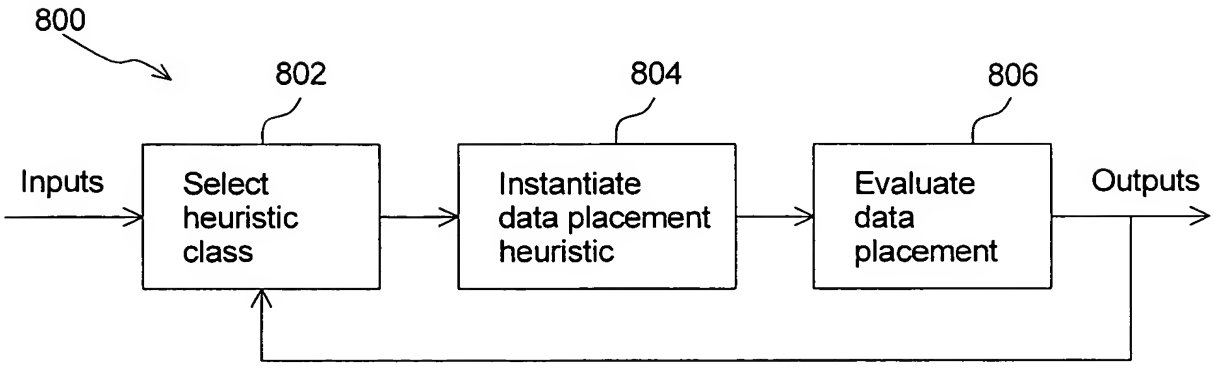
800
802
804
806
Inputs
Select heuristic class
Instantiate data placement heuristic
Evaluate data placement
Outputs

FIG. 8

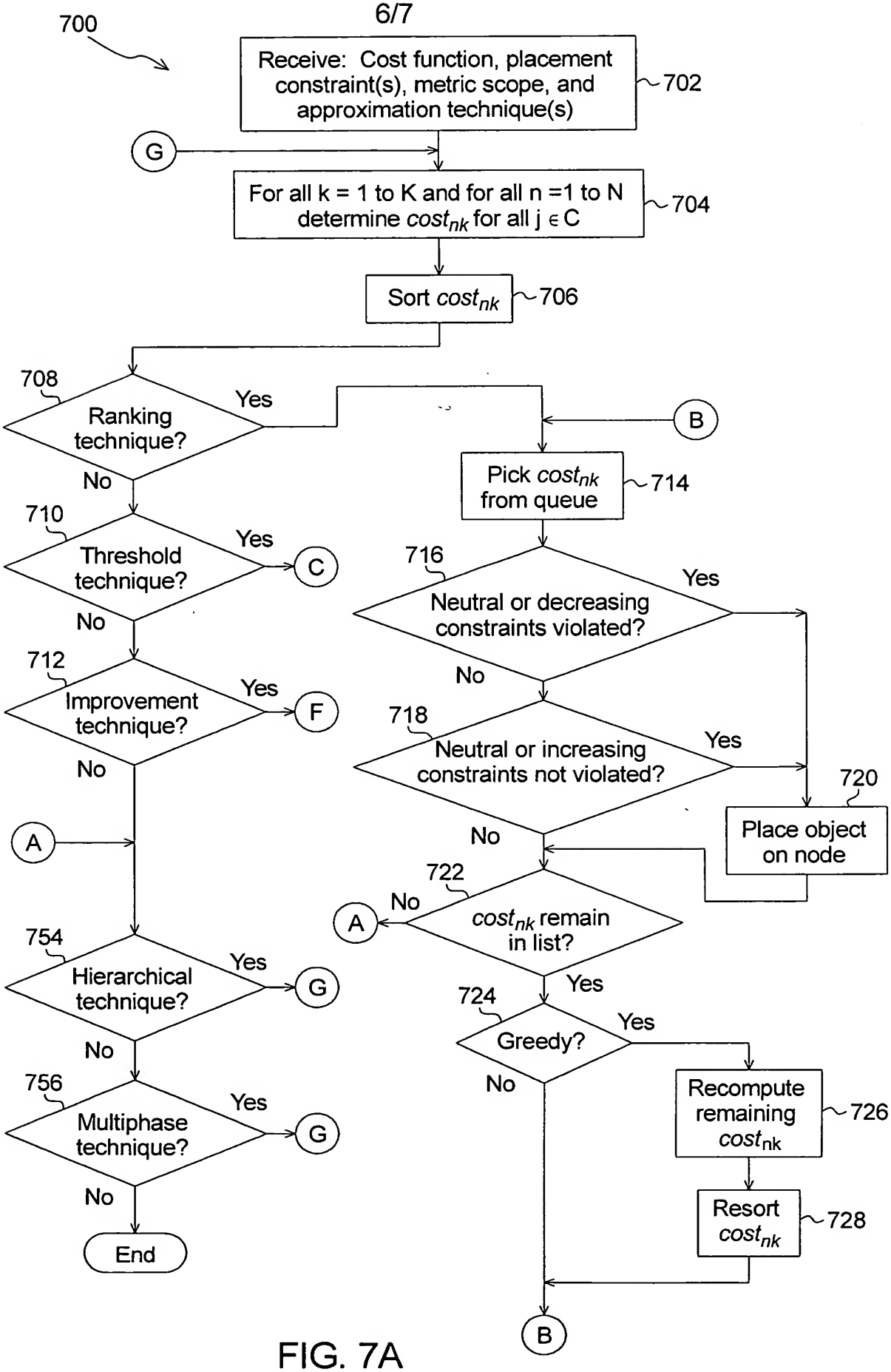
700
Receive: Cost function, placement constraint(s), metric scope, and approximation technique(s)
702
G
For all k = 1 to K and for all n =1 to N determine $cost_{nk}$ for all j ∈ C
704
Sort $cost_{nk}$
706
708
Ranking technique?
Yes
No
B
Pick $cost_{nk}$ from queue
714
710
Threshold technique?
Yes
C
No
716
Neutral or decreasing constraints violated?
Yes
No
712
Improvement technique?
Yes
F
No
718
Neutral or increasing constraints not violated?
Yes
720
Place object on node
No
A
722
No
A
$cost_{nk}$ remain in list?
754
Hierarchical technique?
Yes
G
No
Yes
724
Greedy?
Yes
No
756
Multiphase technique?
Yes
G
No
Recompute remaining $cost_{nk}$
726
Resort $cost_{nk}$
728
End
B

FIG. 7A

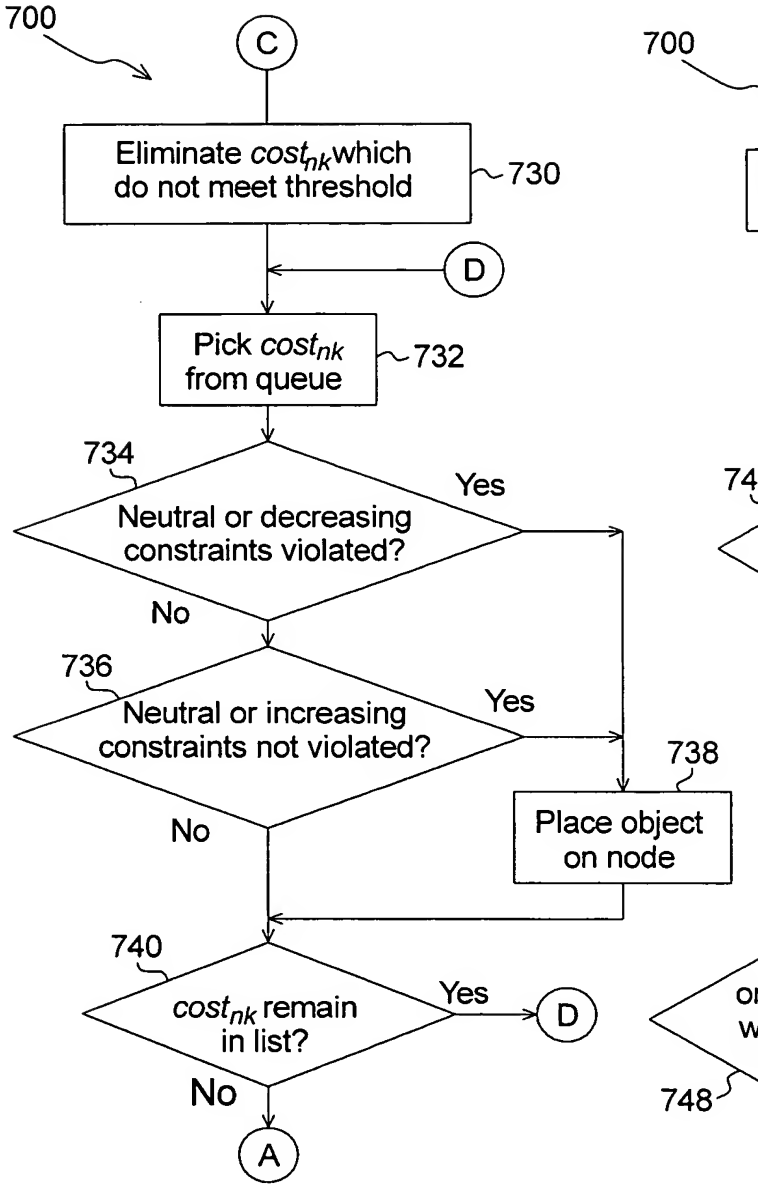
700
C
Eliminate $cost_{nk}$ which do not meet threshold
730
D
Pick $cost_{nk}$ from queue
732
734
Neutral or decreasing constraints violated?
Yes
No
736
Neutral or increasing constraints not violated?
Yes
No
738
Place object on node
740
$cost_{nk}$ remain in list?
Yes
D
No
A

FIG. 7B

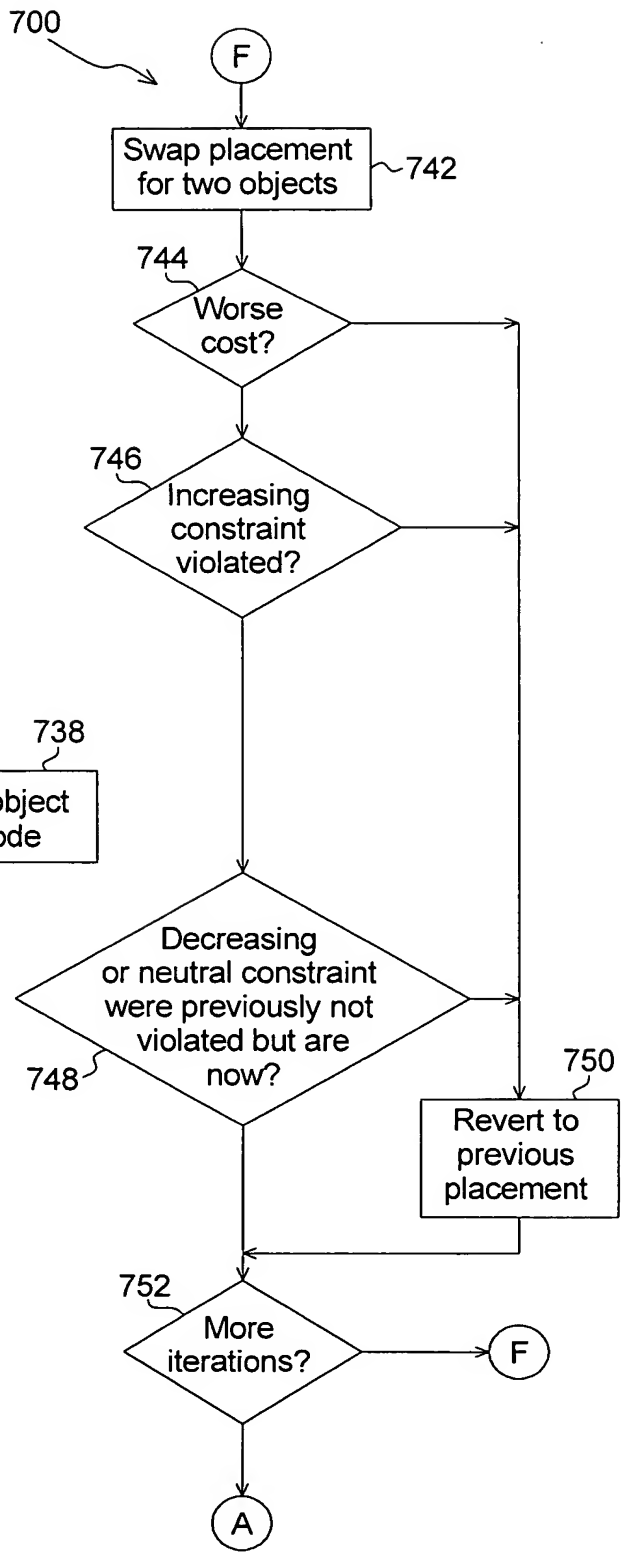
700
F
Swap placement for two objects
742
744
Worse cost?
746
Increasing constraint violated?
Decreasing or neutral constraint were previously not violated but are now?
748
750
Revert to previous placement
752
More iterations?
F
A

FIG. 7C